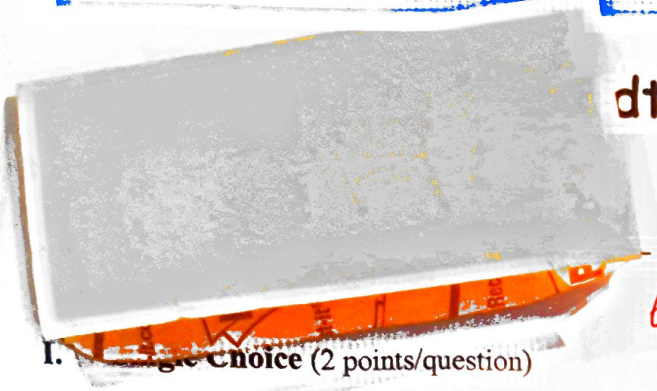
dterm Exam

ID: 1016846631

33/100



I. **Choice** (2 points/question)

1. Which one of the following is not shared by threads?

a) program counter
b) stack
c) both program counter and stack
d) none of the mentioned

Both are per-Thread.

2. The time required to create a new thread in an existing process is

a) greater than the time required to create a new process
b) less than the time required to create a new process
c) equal to the time required to create a new process
d) none of the mentioned

3. Termination of the process terminates

a) first thread of the process
b) first two threads of the process
c) all threads within the process
d) no thread within the process

4. Multithreading an interactive program will improve responsiveness to the user by:

a) continuing to run even if a part of it is blocked
b) waiting for one part to finish before the other begins
c) asking the user to decide the order of multithreading
d) none of the mentioned

5. The kernel is _____ of user threads when the user-level thread management is applied.

master / slave

a) a part of
b) the creator of
c) unaware of
d) aware of

6. If the kernel applies *user level thread management*, then any user level thread performing a blocking system call will:

a) cause the entire process to run along with the other threads
b) cause the thread to block with the other threads of the process running
c) cause the entire process to block even if the other threads of the process are available to run
d) none of the mentioned

7. The following program:

main()

```
{
  if(fork()>0)
  sleep(100);
}
```

results in the creation of:

a) an orphan process

b) a zombie process

c) a process that executes forever

d) none of the mentioned

8. If execve is called immediately after forking,

a) the process specified in the parameter to execve will replace the entire process

b) all the processes will be duplicated

c) all the processes may be duplicated

d) none of the mentioned

9. Which one of the following could be a way of sending a signal to a process?

a) using keyboard to press some special key combinations

b) using shell command to send signal

c) using some system call functions

d) all of the mentioned

10. Which one of the following is NOT a type of operating system?

a) Multiprogramming OS

b) Time sharing OS

c) Batch processing OS.

d) Multiprocessing OS

II. Multiple Choice (4 points/question)

1. Which of the following storage devices can store BIOS programs (i.e., boot codes)

a) RAM/memory

b) ROM

c) Hard disk

d) USB disk

e) EPROM

f) Floppy disk

g) CMOS

2. Which of the following function are considered as system call?

a) fopen()

b) open()

c) printf()

d) write()

e) main()

f) syst_call()

g) fork()

3. Which of the following statements are correct?

a) there could be many waiting queues in the system, but there is only one running queue.

b) applying process context switching is to improve CPU utilization.

c) context switching incurs overhead, which is NOT useful to the system.

d) the most efficient way to kill a zombie process is to kill the zombie process's parent.

e) A parent process and its child process share the same memory space to speed up the process creation time.

f) The fork() function will return 0 to the parent process and PID value to the child process.

g) The wait() function is to suspend a parent process, while one of its child process is running.

4. Which of the following are the methods to achieve inter process communications?

a) shared memory.

b) unnamed pipe.

c) named pipe.

d) TCP/IP.

e) Bluetooth communications.

f) Message caching.

g) Memory passing.

5. Which of the following signals cannot be caught and ignored?

a) SIGABRT.

b) SIGALRM.

c) SIGKILL.

d) SIGTSTP.

e) SIGSTOP

f) SIGPIPE

g) SIGSYS

III. Fill in the blank (2 points/blank)

1. strace command is used to trace system calls made by a process, and ltrace command is used to trace library function calls made by a process.

2. Press "Ctrl-Z" is used to send SIGTSTP signal to the running process.

3. Please fill in the blank in the following program to obtain the user time and system time (***measured in microsecond***) of the "for" loop.

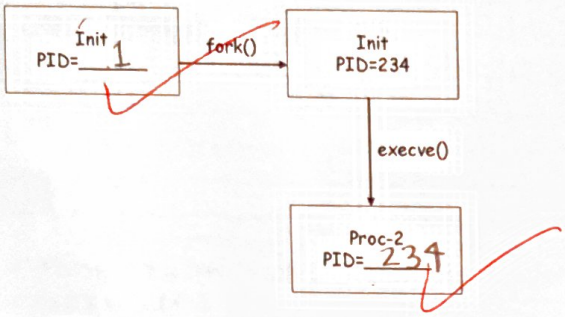
```
int main(int argc, char** argv){
        double dum, usertime, systime;
        struct rusage r;
        struct timeval u_start, s_start, u_end,
s_end;
        getrusage(RUSAGE_SELF, &r);
        u_start=r.ru_utime;
        s_start=r.ru_stime;
    for(int i = 0; i < 100000000; i++)
        {dum=i*exp(0.5)+i;}
        long ru_nvcsw;
        u_end=r.ru_utime;
        s_end=r.ru_stime;
        usertime= struct timeval ru_utime ;
        systime= struct timeval ru_stime ;
        printf("User time: %fus\n", usertime);
        printf("System time: %fus\n", systime);
    return 0;
}
```

4. Master Boot Record (MBR) has three components, i.e., boot loader, Magic #, and Partition Table

5. Please fill in the blank in the following program to achieve pipe communications, where the parent process writes "Hello, midterm exam!" to the pipe, and the child process reads the contents from the pipe.

```
char msg1[20] = "Hello, midterm exam!";
main() {
char inmsg[20]; int pipedes[2]; pid_t pid;
if (pipe(pipedes) < 0)
{ perror("pipe call failure"); exit(1); }
 switch ( case 0 ) {
 case -1: perror("fork call failure"); exit(2);
 case 0:
        pipefd[0] ;
        printf("%s\n",inmsg);break;
 default:
        pipefd[1] ;
        wait(NULL); break;
 }
}
```

6. Please fill in the PID values



+4

IV. General Questions

1. What is the output of the following program being executed? (6p)

0/6

```
int g = 4;  // Global int.

void *myThreadFun(void *varp)
{
   int *myid = (int *)varp;
   static int s = 3;  // Static int
   ++s; ++g;  // increment g=4 & s=3 by 1.
   printf("Thread ID: %d, Static: %d, Global: %d\n", *myid, ++s, ++g);
   return NULL;
}

int main()
{
   int i, j;  // declare values.
   pthread_t tid[3];  // Array TID
   for (i = 0; i < 3; i++) { // for 0,1,2.
      pthread_create(&tid[i], NULL, myThreadFun, (void *)&i);}  //call function.
   for (j=0; j<3; j++) {  // 0, 1, 2
(*)    pthread_join(tid[j], NULL);}
   return 0;
}
```

3 b/c of (*) stuck in for loop.

Output

THREAD ID: 2792, Static: 4, Global: 5

Thread ID: 2793, Static: 5, Global: 6

Thread ID: 2794, Static: 6, Global: 7.

off by 1 b/c of second "++"

2. Assume that there are four processes, and the information of these four processes are as follows:

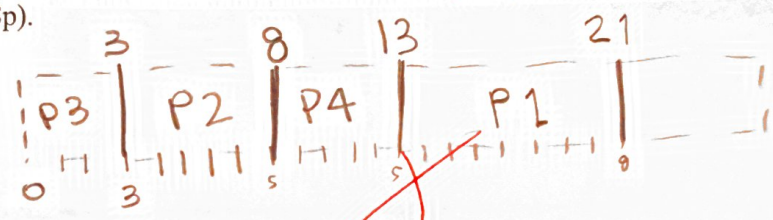
	Arrival time	Service time
P1	0	8
P2	2	5
P3	3	3
P4	5	5

Burst.

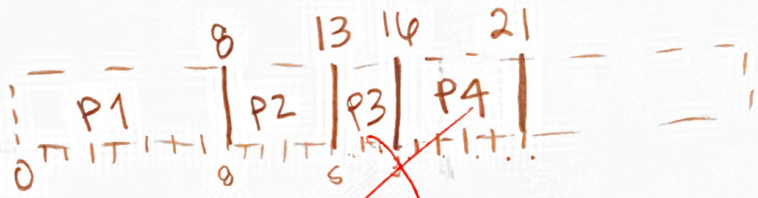
0/16

a) Please draw two Gantt Charts for Shortest Job Next (SJN) and Shortest Remaining-time First (SRF), respectively (8p).

SJN:



SRF



P1, P3, P2, P4

P1, P2, P3, P2, P4, P1

b) Please calculate the average waiting time of a processes for applying SJN and SRF, respectively. Provide one pro and one con of SRF as compared to SJN (8p).

SJN : $\frac{3+8+13+21}{4} = \frac{45}{4} \approx 9.2$

SRF : $\frac{8+13+16+21}{4} = \frac{58}{4} \approx 14.5$

14
4
56

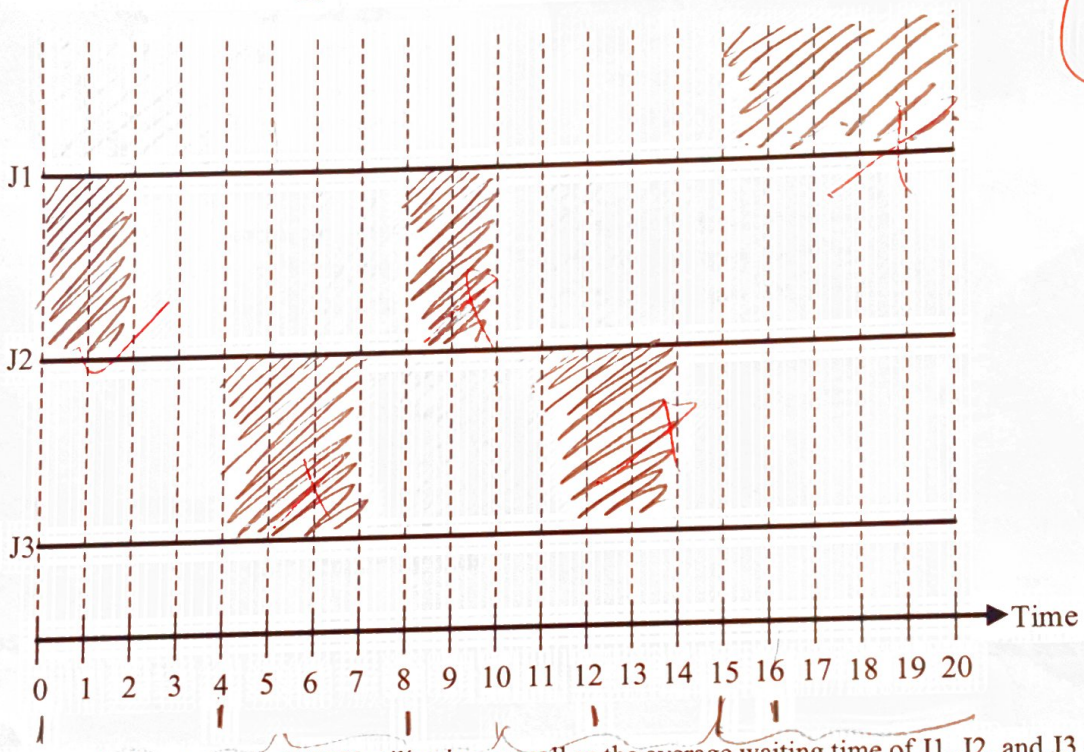
54
16
50

SJN is more efficient than SRF, but SRF leaves less room for errors. X

3. In the single-core real-time OS, ***Earliest Deadline First (EDF)*** is a very popular scheduling algorithm to schedule jobs in the job set. Assume that there are currently three jobs in the job set, and the information of these three jobs are as follows:

	Service time (C)	Relative deadline (D)	Period (T)
J1	5	16	20
J2	2	4	5
J3	3	8	10

a) Please fill out the following job scheduling chart based on the above information (8p):



b) Please calculate the overall CPU utilization as well as the average waiting time of J1, J2, and J3 within 20 time units (8p).

CPU : $5+2+3 = \frac{10}{20} = 50\%$

J1 = $\frac{15}{1}$ = 15 sec

J2 : $\frac{4}{2}$ = 2 sec

J3 : $\frac{6}{2}$ = 3 sec. (J3 is correct) +2

$\frac{0+0+0+2}{4} = 0.5$

16

9

6

Scratch sheet